

COPY

APPLICATION
FOR
UNITED STATES
LETTERS PATENT

SPECIFICATION

For Attorney Docket No. MIT-051 (5473/53)

TO ALL WHOM IT MAY CONCERN:

Be it known that **Craig B. Zilles**, a citizen of the United States of America, residing at 2564 Branch St., #10, Middleton, Wisconsin 53562, in the United States of America; **J. Kenneth Salisbury, Jr.**, a citizen of the United States of America, residing at 20 Pemberton St., Cambridge, Massachusetts 02140, in the United States of America; **Thomas H. Massie**, a citizen of the United States of America, residing at 5 Linlew Dr., Apt. 20, Derry, New Hampshire 03038, in the United States of America; and **David Lawrence Brock**, a citizen of the United States of America, residing at 83 Park Ave., Natick, MA 01760, in the United States of America have invented new and useful improvements in:

METHOD AND APPARATUS FOR DETERMINING FORCES TO BE APPLIED
TO A USER THROUGH A HAPTIC INTERFACE

of which the following is a specification.

PATENT APPLICATION
DOCKET NO.: MIT-051
(5473/53)

**METHOD AND APPARATUS FOR DETERMINING FORCES TO BE APPLIED
TO A USER THROUGH A HAPTIC INTERFACE**

Field of the Invention

The invention relates generally to a method and apparatus for determining forces to be applied to a user interacting with virtual objects in a virtual reality computer environment and more specifically to a method and apparatus for determining forces to be applied to a user through a haptic interface.

Background of the Invention

Virtual reality (VR) computer systems generate simulated environments called "virtual environments" for interaction with a user. The virtual environments include virtual representations of objects which the user can manipulate through an input device. Conventional VR systems attempt to simulate the visual, audio and touch sensory information which would be accessible to a user in the real world environment when interacting with physical objects. These VR systems also attempt to give the user the control over objects that the user would have in the real world environment.

VR system applications include video games, engineering tools and training tools. VR systems have been used to replicate situations which would be too costly or too dangerous to create otherwise. One example of a VR system which is used as a training tool is a flight simulator. Flight simulators replicate cockpits of airplanes and are used to train pilots without subjecting the pilots to the danger of actual flight.

The more sophisticated VR systems include a haptic interface system. A haptic interface system allows a human "observer" to explore and interact with a virtual environment using the sense of touch. The major goal of a haptic interface system is to provide the sensations a user would experience if the user were to touch a virtual

environment. Haptic interface systems replicate the forces felt by humans when interacting with real objects.

The two different forms of human haptic perception that haptic interface systems attempt to replicate are tactile and kinesthetic. The human tactile system consists of
5 nerve endings in the skin which respond to pressure, warmth, cold, pain, vibration and
itch. The tactile system allows humans to sense local geometry, rough texture, and
thermal properties from static contact. The kinesthetic system refers to the collection of
receptors in the muscles, tendons, and joints which allow perception of the motion and
forces upon a human's limbs. In order to accurately replicate the forces experienced by
10 humans in the real world, haptic interface systems attempt to model the shape, surface
compliance and texture of objects.

Haptic interface systems include three main components: a haptic interface
device, a model of the environment to be touched, and a haptic rendering application. A
haptic interface device is a tactile or force-feedback device used by a human which
15 provides the touch sensations of interacting with virtual objects. Known haptic interface
devices consist of an electro-mechanical linkage which can exert a controllable force on
a human's hand. The model of the environment is a computer generated representation
of the real world environment. The haptic rendering application determines the forces to
be applied to the user based on the model environment.

20 One known haptic interface system reduces the user's interactions with the
virtual environment to those of a point interacting with three dimensional objects. The
haptic rendering application used in this known system utilizes vector field methods to
determine the force to be applied to the user. Vector field methods are a classification
for any method that can determine the feedback force to be applied to a user by knowing
25 only the location of the haptic interface point. As used herein, a "haptic interface point"
is defined as the endpoint location of the physical haptic interface as sensed by the
encoders of the VR system. The haptic interface point represents the location of the user
in the virtual environment. Vector field methods however, do not accurately replicate
the touch sensations a user would experience for many objects in the real world. Users

using a haptic interface system which utilizes a vector field method may experience force discontinuities when traversing the volume boundaries of the virtual objects.

Further, vector field methods also do not accurately model thin objects. Due to the limited servo and mechanical stiffnesses, the haptic interface point must travel
5 somewhat into the object before enough force can be applied to the user to make the object feel "solid." When this distance becomes greater than the thickness of an object, the vector field method produces unrealistic sensations. For example, when the haptic interface point penetrates more than halfway through a thin object, rather than exerting a force to push back against the user, the force vector changes direction and applies a force
10 which pushes the user out the side of the object opposite to the side that the user entered. Vector field methods also cannot determine the appropriate forces to apply when the model of the environment overlaps simple objects to create more complex objects.

What is desired then is a haptic interface system which provides touch interfaces which accurately replicate the touch sensations a user would experience in the real
15 world. The present invention permits such functionality.

Summary of the Invention

The invention relates to a method for determining the forces to be applied to a user through a haptic interface. The method includes the steps of generating a
20 representation of an object in graphic space, sensing the position of a user in real space, determining the user's haptic interface in graphic space, determining the user's fiducial object in graphic space and determining a force to be applied to the user in real space. In one embodiment the method calculates a stiffness force to be applied to the user. In other embodiments, the method calculates damping and friction forces to be applied to
25 the user.

In one embodiment, the step of generating a representation of an object in graphic space includes defining the object as a mesh of planar surfaces and associating surface condition values to each of the nodes defining the planar surfaces. In another embodiment, the step of generating a representation of an object in graphic space

includes describing the surface of the object using a coordinate system and associating surface condition values with each set of coordinates.

The invention also relates to an apparatus for determining the forces to be applied to a user through a haptic interface. The apparatus includes a position sensor, a processor executing an algorithm to determine the forces to be applied to a user in real space, a display processor and a force actuator. In one embodiment, the algorithm determining the forces to be applied to the user includes a module generating a representation of an object in graphic space, a module determining the user's haptic interface in graphic space, a module determining the user's fiducial object in graphic space and a module calculating the force to be applied to the user in real space.

The present invention has the technical advantage of accurately replicating the touch sensations a user would experience when interacting with real world objects. The present invention has the further advantage of accurately modeling the forces applied to a user by thin and arbitrarily shaped polyhedral objects. The present invention has yet the further advantage of determining the appropriate forces to be applied to a user by a complex virtual object formed from overlapped simple virtual objects.

Brief Description of the Drawings

Fig. 1 is a flowchart representation of an embodiment of a process for determining a force to be applied to a user through a haptic interface;

Fig. 2 is a flowchart representation of an embodiment of a process for determining a feedback force to be applied to a user through a haptic interface;

Fig. 3 is a pictorial view of a representation of a real world object in graphic space;

Fig. 4A is a pictorial view of a convex portion of a virtual object formed by two planar surfaces and a fiducial object located on one of the surfaces;

Fig. 4B is a pictorial view of the two planar surfaces of Fig. 4A and the fiducial object of Fig. 4A transitioning between the two planar surfaces;

Fig. 4C is a pictorial view of the two planar surfaces of Fig. 4A and the fiducial object of Fig. 4A after the fiducial object has transitioned between the surfaces;

Fig. 5A is a pictorial view of a concave portion of a virtual object formed by two planar surfaces and a fiducial object located on one of the surfaces;

5 Fig. 5B is a pictorial view of the two planar surfaces of Fig. 5A after the fiducial object has penetrated one of the surfaces;

Fig. 6A is a perspective view of a complex virtual object formed from two simpler virtual objects;

10 Fig. 6B is a cross-sectional view of the complex virtual object of Fig. 6A taken through line A-A' of Fig. 6A;

Fig. 7 is a flowchart representation of an embodiment of a process for removing hidden surfaces of complex virtual objects;

Fig. 8 is a flowchart representation of an embodiment of a process for determining a friction force to be applied to a user through a haptic interface;

15 Fig. 9 is a graphical representation of a friction force applied to a user to model friction with slip;

Fig. 10 is a pictorial view of one of the triangular planar surfaces forming the surface of a virtual object;

20 Fig. 11 is a flowchart representation of an embodiment of a process for performing surface smoothing of a virtual object;

Fig. 12 is a flowchart representation of another embodiment of a process for performing surface smoothing of a virtual object;

Fig. 13 is a flowchart representation of embodiment of a process for modeling texture on the surface of a virtual object;

25 Fig. 14A is a pictorial view of one of the planar surfaces forming the surface of a virtual object;

Fig. 14B is a pictorial view of the texture map to be mapped onto the planar surface of Fig. 14A; and

Fig. 15 is a flow diagram of one embodiment of the invention.

Like reference characters in the respective drawn figures indicate corresponding parts.

Detailed Description of the Invention

5 In brief overview, and referring to Fig. 1, a flowchart shows the steps performed by one embodiment of the method of the present invention for determining the forces to be applied to a user through a haptic interface device. In step 10, the haptic rendering application generates a representation of a real world object in graphic space. As used herein, "rendering" is defined as the creation of an image in graphic space. "Haptic rendering application" refers to the application which generates the representation of the
10 real world object and determines the forces to be applied to the user through the haptic interface. As used herein, "graphic space" is defined as the computer generated virtual environment with which the user can interact. In one embodiment, the haptic rendering application uses mathematical models to create the representation of the object. In
15 another embodiment, a separate application is used to create the representation of the object. For example, in one embodiment, a Computer-aided design (CAD) software application is used to generate the representation of the object. The real world objects capable of being represented include planar surfaces, curved surfaces and arbitrarily shaped polyhedral objects. The real world objects may also include concave, convex and
20 curved portions. As used herein, "virtual object" is defined as the representation of the real world object in graphic space.

 In step 12, the sensors of the haptic interface system sense the position of the user in real space. As used herein, "real space" is defined as the real world environment. In step 14, the haptic rendering application utilizes the information obtained by the
25 sensors to determine the haptic interface in graphic space. The location of the haptic interface describes the position of the user in the virtual environment. In step 16, the haptic rendering application determines the fiducial object in graphic space. The fiducial object is the "virtual" location of the haptic interface. The fiducial object location represents the location in graphic space at which the haptic interface would be located if

In more detail and referring now to Fig. 2, a flowchart illustrates a more detailed sequence of steps performed by one embodiment of the present invention to determine a feedback force to be applied to a user in real space through a haptic interface. In the embodiment illustrated by the flowchart of Fig. 2, the user's interactions with the virtual environment are reduced to those of a point interacting with three dimensional objects. In other embodiments, the user's interactions are not reduced to those of a point interacting with three dimensional objects. In other embodiments, the haptic interface and the fiducial object may be a series of points. In still other embodiments, the haptic interface and fiducial object may be three-dimensional objects.

In step 20, the haptic rendering application generates a representation of a real world object in graphic space. As described above, this representation is termed the virtual object. The real world objects modeled by the method of the present invention may have concave portions as well as convex portions. Many different methods can be used to generate the virtual object. In one embodiment, the haptic rendering application defines the real world object as a mesh of planar surfaces. In one embodiment utilizing the mesh of planar surfaces method, each of the planar surfaces comprising the mesh has the same number of sides and the same number of nodes. In another embodiment, the planar surfaces comprising the mesh have varying numbers of sides and nodes. In the preferred embodiment, each of the planar surfaces is triangular and has three nodes. In another embodiment, the haptic rendering application defines the real world object as an n-noded polygon. In still another embodiment, the haptic rendering application describes the real world object using a coordinate system. In yet another embodiment, the representation of the object is displayed on a display.

Fig. 3 shows an example of a representation of a real world object which has been generated by one embodiment of the present invention. The real world object depicted in Fig. 3 is a space shuttle. The representation consists of 616 polygons. In one embodiment, the representation is generated from a standard object file format such as AutoCad's DXF or WAVEFRONT's OBJ.

Once the haptic rendering application has determined the locations of the haptic interface point and the fiducial object point, in step 28 the haptic rendering application calculates the stiffness force component of the feedback force to be applied to a user in real space through the haptic interface. The stiffness force represents the force that would be applied to the user in the real world by a real world object due to the stiffness of the surface of the object. Simple impedance control techniques can be used to calculate the stiffness force to be applied. In one embodiment, the haptic rendering application uses Hooke's law to calculate the stiffness force as illustrated by equation (1) below, wherein k is the stiffness of the virtual object's surface.

10 (1)
$$F_{stiffness} = k (x_{fiducial-object} - x_{haptic-interface})$$

In equation (1), $F_{stiffness}$ represents the stiffness force to be applied to the user through the haptic interface, $x_{fiducial-object}$ represents the position of the fiducial object in graphic space, $x_{haptic-interface}$ represents the position of the haptic interface in graphic space and k represents the stiffness of the virtual object's surface. As shown by equation (1), to calculate the stiffness force, the haptic rendering application first calculates the displacement between the fiducial object point location and the haptic interface point location, represented in equation (1) by $(x_{fiducial-object} - x_{haptic-interface})$. The haptic rendering application then multiplies this displacement by the stiffness of the virtual object's surface, k .

After determining the locations of the haptic interface point and the fiducial object point, the haptic rendering application stores state variables representing these locations for later use in calculating the forces to be applied to the user. The purpose of storing information relating to these locations is to enable the haptic rendering application to compute the forces to be applied to the user based on the history of the user's motions.

In order to accurately model the forces that would be exerted on a user in the real world, in one embodiment, the haptic interface application adds a damping force to

the stiffness force calculated in step 28. The combination of a stiffness force and a damping force provides a more accurate model of the local material properties of the surface of an object.

To obtain the necessary information, the haptic rendering application next
5 determines the velocity of the haptic interface point in step 30 and determines the velocity of the fiducial object point in step 32. In one embodiment, the haptic rendering application determines the velocities of the haptic interface point and the fiducial object point relative to a common reference. The common reference may be a virtual object or simply a point in the virtual environment. In another embodiment, the haptic rendering
10 application determines the velocity of the fiducial object point relative to the haptic interface point. After the haptic rendering application has determined the velocities of the haptic interface point and the fiducial object point in steps 30 and 32, it calculates a damping force to be applied to the user in real space as illustrated by step 34.

In one embodiment of the method of the present invention, the damping force
15 ($F_{damping}$) is based on the motion of the haptic interface point ($\dot{x}_{haptic-interface}$) relative to the motion of the fiducial object point ($\dot{x}_{fiducial-object}$). In another embodiment, only motion in a direction normal (\hat{N}) to the surface of the virtual object is used to calculate the damping force so that motion of the user tangential to the surface of the virtual object is not impeded. In one embodiment, the haptic rendering application computes the
20 damping force according to equation (2) in which c is the damping coefficient and \hat{N} represents the vector normal to the surface of the virtual object.

$$(2) \quad F_{damping} = c \left((\dot{x}_{fiducial-object} - \dot{x}_{haptic-interface}) \cdot \hat{N} \right) \hat{N}$$

In one embodiment, the haptic rendering system only applies a damping force to
25 the user when the calculated damping force has the effect of stiffening the virtual object's surface. The purpose of only applying a damping force which has the effect of stiffening the surface is to avoid the surface having the effect of resisting the withdrawal

of the haptic interface from the surface. This embodiment would not exert a force against the user that would inhibit the user from moving away from the virtual object's surface. Otherwise, the damping force would make the object feel sticky to the user.

Once the haptic rendering application calculates the stiffness and damping forces
5 to be applied to the user in real space, in step 36 the haptic rendering application calculates a feedback force ($F_{feedback}$) to be applied to the user by summing the stiffness ($F_{stiffness}$) and damping ($F_{damping}$) forces as shown by equation (3) below.

$$(3) \quad F_{feedback} = F_{stiffness} + F_{damping}$$

As described above in the discussion of Fig. 3, one embodiment of the method of
10 the present invention generates the representation of the real world object in graphic space by describing the object as a mesh of planar surfaces. Simulating real world objects using surfaces provides an accurate model for the user to interact with because in the real world humans interact with objects on the objects' surfaces. Also as discussed above, one embodiment of the method of the present invention generates the
15 representation of the real world object by defining the object as a mesh of triangular planar surfaces. This embodiment uses a mesh of triangular elements because this representation is the most fundamental, and assures that all of the nodes of each surface are coplanar. Because graphic models do not require the exactness required by haptic models, it is not uncommon to find graphic representations of objects with four-noded
20 surfaces where the four nodes are not coplanar. When the nodes are not coplanar, the fiducial object point may slide between two surfaces and no longer remain on the surface of the virtual object. This would cause the haptic rendering application to calculate incorrect forces to be applied to the user. The problems caused by such surfaces can be avoided by using a triangular mesh. In addition, since three points define a plane, the
25 nodes defining the virtual object can be moved at any time and the object will still be composed of surfaces that are geometrically acceptable for calculating forces according to the method of the present invention.

Another embodiment of the invention takes advantage of this ability to move nodes to implement representations of objects having deformable surfaces. This embodiment simulates deformable surfaces by moving the nodes defining the virtual object in response to forces applied to the deformable surfaces by the user.

5 Figs. 4A-4C illustrate the steps performed by one embodiment of a haptic rendering application to move a fiducial object point 38 between two planar surfaces, 40 and 42 respectively, joined to form a convex surface, when the haptic interface point 44 moves past the intersection 46 of the two planar surfaces 40 and 42. The two planar surfaces 40 and 42 act as constraints on the motion of the fiducial object point 38. That is, the fiducial object point 38 can not penetrate the surfaces 40 and 42. In order to determine the new location of the fiducial object point 38 in response to a change in location of the haptic interface point 44, one embodiment of the method of the present invention first determines the active constraints on the motion of the fiducial object point 38. For infinite planar surfaces, the haptic rendering application of this embodiment 15 defines the planar surface as an active constraint if the fiducial object point 38 is located a positive distance from the planar surface 40 or 42 and the haptic interface point 44 is located a negative distance from the planar surface 40 or 42. The distance is positive if the point is located in the direction of the surface normal pointing outward from the surface of the virtual object. The distance is negative if the point is located in the direction of the surface normal pointing inward from the surface of the virtual object. 20 Using this definition of an active constraint causes the virtual surfaces 40 and 42 to act as one-way constraints to penetration. That is, the surfaces 40 and 42 only prevent the fiducial object point 38 from entering the surfaces 40 and 42.

For surfaces that are not of infinite extent, in addition to the requirements for 25 infinite planar surfaces, to be defined as an active constraint the haptic rendering application requires that the contact of the fiducial object point 38 with the plane containing the surface take place within the boundaries of the planar surface. In order to determine whether the contact of the fiducial object point 38 takes place within the boundaries of the planar surface, in one embodiment the haptic rendering application

determines the line intersecting the current haptic interface point 44 and the old fiducial object point 38 which the haptic rendering application is updating. If this line passes through the planar surface within the boundaries of the surface, then the haptic rendering application defines the surface as active.

5 Figs. 4A-4C show two surfaces, 40 and 42 respectively, of a convex portion of a virtual object. In the embodiment wherein the user's interactions with the virtual environment are reduced to those of a point interacting with three dimensional objects, when a user interacts with convex portions of objects, only one surface of the virtual object is an active constraint at a time. To transition the fiducial object point 38 between
10 two surfaces sharing a convex edge requires two steps. In Fig. 4A, surface 40 is an active constraint because the fiducial object point 38 is located a positive distance from the surface 40, the haptic interface point 44 is located a negative distance from the surface 40 and the fiducial object point 38 is located within the boundaries of the surface 40. While surface 40 remains an active constraint, the fiducial object point 38 remains
15 on the plane 48 containing the surface 40, but not does not necessarily remain within the physical boundaries of surface 40.

Fig. 4B illustrates the first step performed by the haptic rendering application to transition the fiducial object point 38 to the surface 42. In the first step, the haptic rendering application moves the fiducial object point 38 over the second surface 42, but
20 the fiducial object point 38 remains in the plane 48 of the first surface 40. In the second step, the haptic rendering application no longer considers the first surface 40 an active constraint because the fiducial object point 38 is not located within the physical boundaries of the surface 40. The haptic rendering application then moves the fiducial object point 38 onto the second surface 42 as illustrated in Fig. 4C. In other
25 embodiments wherein the user's interactions with the virtual environment are not reduced to those of a point interacting with three dimensional objects, more than one surface of a convex portion of a virtual object may be an active constraint at a time.

Figs. 5A and 5B show a concave portion of an object defined by two planar surfaces, 50 and 52 respectively. When the user interacts with a concave portion of an

object, multiple surfaces can be active simultaneously. When the user interacts with the concave intersection 53 of the two planes 55 and 57, the haptic rendering application defines both planes, 55 and 57 respectively, as active constraints and the motion of the fiducial object 54 is restricted to the line defined by the intersection of the two planes 55 and 57. When the user is in contact with the intersection of three planar surfaces, all three surfaces are active constraints and the fiducial object is confined to the point defined by the intersection of the three surfaces. At the intersection of more than three surfaces, the haptic rendering application considers only three of the surfaces as active constraints at any one time and the fiducial object point is again confined to a point.

Figs. 5A and 5B illustrate the situation that occurs when the user interacts with surfaces that intersect at an acute angle. Fig. 5A shows the location of the fiducial object point 54 when the user presses into the surface 50 to the location defined by the haptic interface point 56. As the user slides down along the surface 50, the fiducial object point 54 may cross over the surface 52 before the surface 52 meets the requirements outlined above to be defined as an active constraint.

Fig. 5B illustrates the situation in which the fiducial object point 54' has crossed over surface 52 before the surface 52 is considered an active constraint. According to the definition of an active constraint discussed above, surface 52 does not qualify as an active constraint in Fig. 5B because the haptic interface point 56' is not a negative distance from the plane 57 containing the surface 52. To solve this problem, in one embodiment, the haptic rendering application iterates the process of determining the new fiducial object point location. During the first iteration, the haptic rendering application determines a set of active constraints and calculates the new fiducial object point location 54'. During the second iteration the haptic rendering application uses the "new" fiducial object point location 54' as the haptic interface point location in combination with the "old" fiducial object point location 54 to check the neighboring surfaces to determine whether any additional surfaces qualify as active constraints. If the haptic rendering application determines that an additional surface qualifies as an active

constraint, the haptic rendering application continues the iterations and updates the fiducial object point location until no new active constraints are found.

In one embodiment, once the complete set of active constraints is found, the haptic rendering application uses Lagrange multipliers to update the location of the fiducial object point. Lagrange multipliers are used in maximizing or minimizing functions which have several variables which are subject to constraints. In this embodiment, the virtual environment is described by a rectangular coordinate system having coordinate sets with three entries. The haptic rendering application uses equation (4) below to model the energy of a virtual spring of unity stiffness. In equation (4), Q represents the energy in a virtual spring between the fiducial object and the haptic interface, x , y and z represent the coordinates of the fiducial object point and x_p , y_p and z_p represent the coordinates of the haptic interface point. In equation (4), the spring constant equals 1. The goal in solving equation (4) is to minimize the value of Q , thereby making the virtual spring as small as possible.

$$(4) \quad Q = \frac{1}{2}(x - x_p)^2 + \frac{1}{2}(y - y_p)^2 + \frac{1}{2}(z - z_p)^2$$

The haptic rendering application then adds the active constraints as planes according to equation (5). In equation (5), A_n , B_n and C_n indicate the direction of the surface normal to the plane containing the constraint. D_n indicates the distance from the origin of the plane containing the active constraint.

$$(5) \quad A_n x + B_n y + C_n z - D_n = 0$$

The first step in utilizing Lagrange multipliers is to form a function L of the variables in the equation to be minimized and the equations defining the constraints. In the case of

three constraints, L will be a function of x, y, z, l_1, l_2 and l_3 , where l_1, l_2 and l_3 are the Lagrange multipliers. The function L will be in the form of:

$$L(x, y, z, l_1, l_2, l_3) = (\text{function to be minimized}) - l_1(\text{constraint}_1) - l_2(\text{constraint}_2) - l_3(\text{constraint}_3).$$

5

Following this model, the haptic rendering application combines equations (4) and (5) to generate equation (6).

$$(6) \quad L = \frac{1}{2}(x - x_p)^2 + \frac{1}{2}(y - y_p)^2 + \frac{1}{2}(z - z_p)^2 \\ + l_1(A_1x + B_1y + C_1z - D_1) \\ + l_2(A_2x + B_2y + C_2z - D_2) \\ + l_3(A_3x + B_3y + C_3z - D_3)$$

10

The haptic rendering application calculates the new location of the fiducial object point by minimizing L in equation (6). To minimize L , the haptic rendering application first computes the six partial derivatives of equation (6). The haptic rendering application then minimizes L by setting all six partial derivatives of L to 0. This results in six simultaneous equations with six variables (x, y, z, l_1, l_2 and l_3) to solve for. The six partial derivative equations can be organized into a set of simultaneous equations represented by the matrix equation (7) below.

20

$$(7) \quad \begin{bmatrix} 1 & 0 & 0 & A_1 & A_2 & A_3 \\ 0 & 1 & 0 & B_1 & B_2 & B_3 \\ 0 & 0 & 1 & C_1 & C_2 & C_3 \\ A_1 & B_1 & C_1 & 0 & 0 & 0 \\ A_2 & B_2 & C_2 & 0 & 0 & 0 \\ A_3 & B_3 & C_3 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ l_1 \\ l_2 \\ l_3 \end{bmatrix} = \begin{bmatrix} x_p \\ y_p \\ z_p \\ D_1 \\ D_2 \\ D_3 \end{bmatrix}$$

The matrix equation (7) has a number of useful properties. It is symmetric, the upper left hand corner (3 x 3) is always the identity matrix, the lower left hand corner is always a null matrix, and the matrix is invertible. Solving the matrix equation (7) also does not require row swapping. Because of these properties, x , y and z can be solved for
5 in only 65 multiplicative operations. In the case when there are only two active constraints, the leftmost matrix is a (5 x 5) matrix and x , y and z can be solved for in only 33 multiplicative operations. In the single constraint case, the leftmost matrix is a (4 x 4) matrix and x , y and z can be solved for in only 12 multiplicative operations. As described above, when there are no active constraints, the fiducial object point is located
10 at the position of the haptic interface point and no calculations are required.

Figs. 6A and 6B show an example of a complex object 58 formed from the overlapping of two simple objects, 60 and 62 respectively. As described above, one of the problems with existing haptic rendering applications is that they can not determine the appropriate forces to apply to a user when the model of the environment overlaps
15 simple objects to create more complex objects. When multiple objects are in close proximity, a naive active constraint detector will return too many active surfaces and the computed force will be incorrect. One embodiment of the method of the present invention includes a method for removing hidden surfaces to assure that the fiducial object point is located on the appropriate surface of the virtual object. A hidden surface
20 is a surface of a virtual object which is covered by a surface of another virtual object.

In the example shown in Figs. 6A and 6B, when a user presses down on surface 64, the haptic interface point 66 penetrates both surface 64 and surface 68. Without removing the hidden surface 69, the haptic rendering application would define both surfaces 64, 68 as active constraints and would not calculate the location of the fiducial
25 object point 70 correctly.

Fig. 7 shows a flowchart illustrating the steps performed by one embodiment of the present invention to remove hidden surfaces. In step 71, the haptic rendering application determines the surfaces that qualify as active constraints. In step 72, the haptic rendering application determines the redundant surfaces. Redundant surfaces are

surfaces which have been defined as active constraints and which have the same or similar surface normals. Referring again to Figs. 6A and 6B, surfaces 64 and 68 are redundant surfaces. Referring back to Fig. 8, in step 74, the haptic rendering application determines the redundant surface which is closest to the fiducial object and add this

5 surface to the list of non-redundant active constraints. In step 76, the haptic rendering application orders the list of non-redundant active constraints by the distance of the active constraints to the fiducial object. Next, in step 78, the haptic rendering application computes the new fiducial object location using the closest non-redundant active constraint in equations (4)-(7) above. In step 80, the haptic rendering application

10 determines whether the updated fiducial object location is valid. The updated fiducial object location is valid unless it penetrates a surface of the virtual object.

If the fiducial object location is valid, in step 82 the haptic rendering application waits for a change in position of the haptic interface point before repeating the process. If the fiducial object location is not valid, in step 84 the haptic rendering application

15 adds the crossed active constraint which is nearest to the fiducial object point to the computation matrix of equation (7). Next, in step 86 the haptic rendering application recomputes the fiducial object location. The haptic rendering application then repeats steps 80, 84 and 86 until the computed fiducial object point location is valid.

When humans interact with objects in the real world, the objects exert both a

20 normal force and a friction force to the human. In order to accurately model real world interactions with an object, one embodiment of the method of the present invention includes a method for determining and applying friction forces to a user.

Fig. 8 shows a flowchart of the steps performed by one embodiment of the present invention for determining the friction forces to be applied to a user in the real

25 world through a haptic interface device. The method illustrated by the flowchart in Fig. 8 determines the static friction force to be applied to the user in real space. Friction models with stiction in general have two states of contact: stiction and kinetic friction. In the stiction state the user has made contact with an object but has not provided enough tangential force to "break away." When enough force is applied, a transition is made to

the kinetic friction state. In the kinetic friction state a force is applied in a direction to impose the direction of motion. All friction forces are applied to the user in directions tangential to the feedback force discussed above.

In step 88, the haptic rendering application determines the location of the
5 stiction point in graphic space. The location of the stiction point starts at the location in graphic space where the user first makes contact with the virtual object. The location of the stiction point is in reference to the location of the fiducial object. The location of the stiction point starts at the location where the fiducial object makes contact with the virtual object. Once the stiction point is determined, the haptic rendering application
10 determines the velocity of the stiction point in step 90. Next, in step 92, the haptic rendering application calculates the friction force to be applied to the user in real space.

One embodiment of the present invention uses equations (8), (9) and (10) below to calculate the friction force to be applied to the user. In these equations, $x_{stiction-point}$ is the location of the stiction point, $x_{haptic-interface}$ is the location of the haptic interface and
15 Δx represents the displacement between the stiction point location and the haptic interface point location.

$$(8) \quad \Delta x = (x_{stiction-point} - x_{haptic-interface})$$

20

$$(9) \quad \Delta x_{tangential} = \left(\Delta x - (\Delta x \cdot \hat{N}) \cdot \hat{N} \right)$$

In equation (9), $\Delta x_{tangential}$ represents the component of the displacement between the stiction point location and the haptic interface point location that is tangential to the
25 surface of the virtual object. \hat{N} is a unit vector which is normal to the surface of the virtual object and points outward from the surface. The purpose of equation (9) is to determine the component of the displacement vector between the stiction point location

and the haptic interface point location that is not in the direction normal to the surface of the virtual object.

$$(10) \quad F_{friction} = k\Delta x_{tangential} + c\Delta \dot{x}_{tangential}$$

5

In equation (10), $F_{friction}$ represents the friction force to be applied to the user. k represents the maximum stiffness value of the surface that the haptic interface device can apply to the user without becoming unstable. c represents the maximum viscosity of the virtual object's surface that the haptic interface device can apply to the user without becoming unstable. When the haptic interface device attempts to apply a stiffness value (k) or a viscosity (c) that is greater than the maximum value, the haptic interface device may become unstable. In one embodiment, when the haptic interface device becomes unstable, it begins to vibrate. $\Delta \dot{x}_{tangential}$ represents the rate of change of the component of the displacement vector between the stiction point location and the haptic interface point location that is tangential to the surface of the virtual object.

10
15

After the haptic rendering application has determined the friction force to be applied to the user, one embodiment of the present invention calculates the total force to be applied to the user according to equation (11) below.

20

$$(11) \quad F_{total} = F_{feedback} + F_{friction}$$

In equation (11) the calculated friction force ($F_{friction}$) is added to the calculated feedback force ($F_{feedback}$) to determine the total force (F_{total}) to be applied to the user.

25

Referring again to the flowchart of Fig. 8, once the haptic rendering application has determined a friction force to be applied to a user, the haptic rendering application performs a series of steps according to equation (12) below to update the stiction point location and the friction force to be applied to the user.

$$(12) \quad F_{friction} \geq \mu F_{feedback}$$

In equation (12) $F_{feedback}$ represents the sum of the stiffness force and the damping force calculated above. In step 94, the haptic rendering application multiplies the sum of the stiffness force and the damping force ($F_{feedback}$) by a coefficient of friction μ to obtain a product. In step 96, the haptic rendering application compares this product to the calculated friction force to determine whether the calculated friction force is greater than or equal to the product. The purpose of equation (12) is to determine whether the calculated friction force can be applied to the user without violating the laws of physics. If the friction force is too large and violates the laws of physics, the stiction point must be updated until a friction force is calculated that can be applied to the user without violating the laws of physics. If the friction force is less than the product, the haptic rendering application proceeds to step 98 and adds the friction force to the sum of the stiffness force and the damping force to calculate a total force to be applied to the user in real space through a haptic interface device. If the calculated friction force is greater than or equal to the product, the haptic rendering application proceeds to step 100 and updates the stiction point location according to equation (13) which will be discussed below. Next, in step 102 the haptic rendering application recalculates the friction force to be applied to the user and returns to step 96. The haptic rendering application repeats steps 96, 100 and 102 until the friction force is less than the product obtained in step 94.

In step 100 the haptic rendering application updates the position of the stiction point. When the old stiction point is broken, the new stiction point location is calculated using the new fiducial point location and the old stiction point location. The new stiction point is placed on the line intersecting the new fiducial object point and the old stiction point. The new stiction point is placed at a distance ($\Delta x_{tangential}$) from the new fiducial object point so that the force on the user is equal to the maximum friction force, as described by equation (13). In equation (13), c is the viscosity, μ is the coefficient of

friction, k is the stiffness value of the surface and F_{normal} is the calculated normal force. $\Delta \dot{x}_{tangential}$ represents the rate of change of the component of the displacement vector between the stiction point location and the haptic interface point location that is tangential to the surface of the virtual object.

5

$$(13) \quad \Delta x_{tangential} = \frac{\mu}{k} F_{normal} - \frac{c}{k} \Delta \dot{x}_{tangential}$$

In the real world there is an additional slipping sensation which a user experiences when the user is slipping relative to an object. Also, in the real world there is usually some vibration associated with slipping. In order to model the sensation of slipping, one embodiment of the method of the present invention utilizes two coefficients of friction. This embodiment uses one coefficient of friction to determine whether the stiction point location should be updated, and another slightly lower coefficient of friction for calculating the new location of the stiction point. The result of this method is that each time a new stiction point is placed, the friction force is lower, and a small distance must be traveled for the user to break away again.

Fig. 9 shows the saw-tooth wave that results when a constant relative velocity is specified between the fiducial object and the virtual object being touched. When the stiction point is updated, the new position is calculated using a coefficient of friction that is smaller than the one used to determine whether the maximum friction force has been exceeded. When a constant velocity is specified a vibration is felt due to the saw-tooth like variation in friction force.

One embodiment of the method of the present invention uses equations (14) and (15) below to model friction with slip. As discussed above, since the fiducial object can not penetrate the surface of a virtual object, when the haptic interface has penetrated the surface of a virtual object, the fiducial object remains on the surface of the virtual object. Therefore, as long as the fiducial object remains on the surface of the virtual object there

is no non-tangential motion of the fiducial object with respect to the surface of the virtual object and equation (9) above can be simplified to equation (14) below.

$$(14) \quad \Delta x_{\text{tangential}} = (x_{\text{stiction-point}} - x_{\text{fiducial-object}})$$

5

$\Delta x_{\text{tangential}}$ represents the component of the displacement between the stiction point location ($X_{\text{stiction-point}}$) and the fiducial object point location ($X_{\text{fiducial-object}}$) that is tangential to the surface of the virtual object.

10 If the feedback force calculated above is zero (0) then the stiction point should be collocated with the fiducial object point. The position of the stiction point should only be updated when the distance from the fiducial object point to the stiction point exceeds the distance from the haptic interface point to the fiducial object point multiplied by the coefficient of friction (μ). The stiction point location ($X_{\text{stiction-point}}$) can then be used in combination with the haptic interface point location ($X_{\text{haptic-interface}}$) and
15 the stiffness value (k) of the surface to calculate the change in force (ΔF_{total}) to be applied to the user according to equation (15) below.

$$(15) \quad \Delta F_{\text{total}} = k (x_{\text{stiction-point}} - x_{\text{haptic-interface}})$$

20 In yet another embodiment of the present invention, the haptic rendering application performs surface smoothing of a virtual object. In the real world, it is common for the properties of an object to vary across its surface. Therefore, to provide a comparable experience to the user of a haptic interface system, the present invention provides a haptic rendering application for providing to the user tactile feedback of the
25 properties of the surface of an object being touched.

In one embodiment, the present invention provides such feedback to the user by dividing the surface of an object into a mesh of planar surfaces, assigning values to nodes defined by intersecting points in the planar surfaces, and utilizing such values in

an interpolation scheme. This method of using planar surfaces effectively provides a human with the sensation of feeling surface characteristics, notwithstanding the absence of curved surfaces. Planar surfaces can be used to accurately model an object's surface because of the fact that humans have a rather poor sense of position but are extremely sensitive to discontinuities of force direction. If the force exhibited at points normal to the object are smoothed, then the actual shape of the object need not be as true to its actual shape in the real world to provide the user with an adequate simulation of the object.

Referring to Fig. 10, shown is one of the triangular planar surfaces forming a virtual object simulated by the haptic rendering application of one embodiment of the present invention. In this figure, the planar surface forms a triangle, however planar surfaces of other shapes can be used. Typically, the virtual object comprises a plurality of such planar surfaces, as described above and shown in Fig. 3. The haptic rendering application assigns to the triangle a plurality of nodes shown as A, B, and C. As further described below, by assigning a parameter value representing a surface characteristic to each of the nodes of the polyhedra and interpolating the parameter values between nodal values, parameter values at other points within the triangular surface can be determined. In this manner, the haptic rendering application provides continuity in the direction of the force applied to the user across the area defined by the triangle.

Referring to Fig. 11, a flowchart shows the steps employed by one embodiment of the haptic rendering application of the present invention for performing a method of surface smoothing. In step 104, the haptic rendering application defines an object as a mesh of planar surfaces. Referring again to Fig. 3, the planar surfaces define a plurality of polyhedra, each having a plurality of nodes associated therewith. In step 106, the haptic rendering application assigns a parameter value to each node of each planar surface. In one embodiment, the parameter value is the surface normal. Alternatively, in other embodiments of the invention, the parameter may be a stiffness force or a damping force. In step 108, the haptic rendering application determines on which planar surfaces the fiducial object point is located. As discussed above, the fiducial object point

represents the location in graphic space at which the haptic interface point would be located if the haptic interface point could be prevented from penetrating virtual objects. By determining the planar surfaces on which the fiducial object point is located, the haptic rendering application determines the nodes and corresponding parameter values to be utilized in the interpolation scheme to provide a user with a tactile sensation of the surface characteristics of that point. In step 110, the haptic rendering application computes the parameter value at the location of fiducial object point by interpolating the parameter values associated with the nodes assigned to the planar surfaces on which the fiducial object is located.

- 10 Referring again to Fig 10, the fiducial object point is located at point D. In step 108 as described above, the haptic rendering application determines that the fiducial object lies in the planar surface defined by nodes A, B, and C. After determining the nodes defining the planar surface, the interpolation scheme described in step 110 can be accomplished by projecting the vector \overline{AD} to the line \overline{BC} to determine the point E.
- 15 The parameter value for point E is found by interpolating the parameter values of nodes B and C, and is shown mathematically by equation (16) below.

$$(16) \quad \hat{E} = \left| \frac{\overline{BE}}{\overline{BC}} \right| \hat{B} + \left| \frac{\overline{CE}}{\overline{BC}} \right| \hat{C}$$

- Once the parameter value at point E is determined, the parameter value at fiducial object point D is determined in a similar fashion. The parameter value at point D is determined by interpolating between the points A and E, shown mathematically by equation (17) below.

$$(17) \quad \hat{D} = \left| \frac{\overline{AD}}{\overline{AE}} \right| \hat{A} + \left| \frac{\overline{DE}}{\overline{AE}} \right| \hat{E}$$

- 25 Referring to Fig. 12, a flowchart shows the steps employed by another embodiment of the haptic rendering application of the present invention for performing a method of surface smoothing. In this embodiment, the present invention provides tactile

feedback of the surface characteristics of a virtual object to the user through the aid of a coordinate mapping system. Through the use of a coordinate mapping system, the tactile sensation of the varying characteristics of the virtual object can be simulated. Such characteristics can include the texture, as well as the stiffness and damping

5 characteristics of the virtual object. As shown in step 112, the haptic rendering application initially describes the surface of the object using a coordinate system. In one embodiment, the coordinate system is a rectangular coordinate system. In other embodiments, the coordinate system may be a spherical or cylindrical coordinate system. In step 114, each coordinate set comprising the coordinate system is assigned a
10 parameter value representing a characteristic of the virtual object. The characteristic may be a stiffness value, a damping coefficient or a surface normal.

For example, part of the object may be smoother than another portion. To accurately model this, the parameter values at the corresponding coordinate sets would vary accordingly. In step 116, the haptic rendering system determines which set of
15 coordinates describes the location of the fiducial object point. Once this set of coordinates is known, the parameter value representing the surface characteristic at the fiducial object point location is known and can be used to determine a force to be applied to the user representing the texture, stiffness or damping characteristics of the virtual object.

20 In this embodiment of the invention, the force applied to the user is a function of position and changes as the user moves across the surface of the virtual object. Given that the human sensation of texture is carried out by the mechano-receptors in one's finger tip, such changes in forces applied to the user's fingertip adequately simulate such characteristics as texture.

25 Referring to Fig. 13, a flowchart illustrates the steps employed by yet another embodiment of the haptic rendering application of the present invention for performing a method of surface smoothing. In this embodiment, the present invention provides tactile feedback of the surface characteristics of a virtual object to the user through the aid of a coordinate mapping system known as bump mapping. In the graphics world, bump maps

are used to correctly display the illumination of a bumpy surface. Similar to the coordinate mapping system described above, bump maps use coordinate systems to associate a small displacement force to be applied to the user in a direction normal to the surface of the virtual object. The application of a small displacement force models the texture of a surface. For example, as a user moves along a surface, the user can experience the feeling of a bumpy surface.

Referring to step 117, the haptic rendering application initially describes the virtual environment using a coordinate system. The coordinate system can be a rectangular, spherical or cylindrical coordinate system. Referring now to Fig. 14A, a graph shows a planar surface 130 of a virtual object. In Fig. 14A, the haptic rendering application defines the virtual environment, designated generally by reference numeral 132, using a rectangular coordinate system having x , y , and z axes which are orthogonal to each other.

Referring again to Fig. 13, in step 118 the haptic rendering application next describes the surface of the object using a coordinate system. The coordinate system can be a rectangular, spherical or cylindrical coordinate system. In Fig. 14A, the haptic rendering application describes the planar surface 130 using a rectangular coordinate system having s and t axes. In certain embodiments, the s and t axes may not be orthogonal. The coordinates of the (s, t) coordinate system can be mapped into the (x, y, z) coordinate system through a series of transformation equations in which s is a function of x , y and z and t is a function of x , y and z .

Fig. 14B shows the texture map 134 to be applied to the planar surface 130. The texture map 134 is defined by a third coordinate system having u and v axes. The texture map represents the texture to be assigned to the planar surface 130. For example, the bumps 135 illustrate displacement forces to be applied to a user to simulate a bumpy surface. A second series of transformation equations maps the coordinates (s, t) of the planar surface 130 to the equivalent (u, v) coordinates of the texture map 134.

Referring again to Fig. 13, in step 120, each coordinate set (u, v) of the texture map coordinate system is assigned a displacement value to be applied in the direction

normal to the surface of the virtual object. This displacement value represents the force that would be exerted by the surface of the virtual object on the user. The texture function $B(u, v)$ represents the displacement values assigned to the coordinate sets (u, v) . For example, in one embodiment, the texture function $B(u, v)$ represents the height of a bump to be experienced by a user as a function of the user's position in the texture map. As shown in step 122, the haptic rendering application also associates a surface normal for each set of coordinates (s, t) of the planar surface coordinate system.

The embodiment shown in Figs. 4A and 4B utilizes rectangular coordinate systems and the surface of the virtual object 130 has surface coordinates (s, t) and the texture map 134 has coordinates (u, v) . In step 124, the haptic rendering application calculates the new surface normal \bar{N}_{new} for each set of coordinates by adding the displacement force of the texture coordinates (u, v) to the corresponding surface normal of the planar surface coordinates (s, t) as shown below by equation (18).

$$(18) \quad \bar{N}_{new} = \bar{N} + \frac{B_u(\bar{N} \times \bar{P}_t) - B_v(\bar{N} \times \bar{P}_s)}{|\bar{N}|}$$

In equation (18) \bar{N} is the surface normal to the planar surface 130. B_u and B_v are the partial derivatives of the texture function $B(u, v)$ with respect to the u and v directions. B_u represents the curvature of the bump in the u direction and B_v represents the curvature of the bump in the v direction. \bar{P}_s and \bar{P}_t are the partial derivatives of the equation $\bar{P} = [x(s, t), y(s, t), z(s, t)]$ in the s and t directions. \bar{P}_s and \bar{P}_t represent the unit vectors in the planar surface coordinate system illustrated in Fig. 14A.

Associating an additional displacement force to certain coordinate sets replicates where a bump is to appear on a virtual object's surface and be sensed by the user's hand. As shown in step 126, the haptic rendering application then determines which coordinate set describes the fiducial object point. In step 128, the haptic rendering application determines the total surface normal corresponding to that coordinate set and uses that total surface normal to determine the appropriate force to be administered to the user.

The haptic rendering application thereby represents the existence of a bump on the surface of the virtual object to the user.

This embodiment of the invention can be used to simulate materials like wood, sand paper, or rusted metal, for example.

5 The present invention also relates to an apparatus for determining forces to be applied to a user through a haptic interface. Fig. 15 shows an embodiment of an apparatus for determining forces to be applied to a user through a haptic interface. The apparatus includes a sensor 140, a haptic rendering processor 142 for determining the forces to be applied to the user, a display processor 144, a force actuator 148 and a
10 display 150. The purpose of sensor 140 is to sense the position of the user 146. The sensor 140 may be any of the devices known in the art for sensing positions of objects. The haptic rendering processor 142 is in electrical communication with the sensor 140 and executes an algorithm to determine the forces to be applied to the user 146 in real space. The algorithm includes a module generating a representation of a real world
15 object in graphic space, a module determining the user's haptic interface in graphic space, a module determining the user's fiducial object in graphic space and a module calculating a force to be applied to the user in real space. The module determining the user's haptic interface in graphic space translates the position of the user in real space into a position in graphic space. The module determining the user's fiducial object in
20 graphic space determines the location at which the haptic interface would be if the haptic interface could be prevented from penetrating virtual objects. In one embodiment, the user's haptic interface and fiducial object are points in graphic space. In one embodiment, the module calculating a force to be applied to the user in real space calculates a stiffness force to be applied to the user. In other embodiments, this module
25 calculates a damping force, a friction force or a combination of forces to be applied to the user.

The display processor 144 is in electrical communication with the haptic rendering processor 142. The display processor 144 displays the representation of the real world object created by the haptic rendering processor 142 on a display 150. In one

embodiment, the display processor 144 also displays the user's fiducial object location on the display 150. The user's fiducial object location represents the position of the user in graphic space relative to the virtual object. The display 150 may be a computer screen, television screen, or any other device known in the art for displaying images of objects.

- 5 The display 150 may also produce audio sounds in response to the interactions of objects in virtual space.

The force actuator 148 is in electrical communication with the haptic rendering processor 142. The force actuator 148 produces the force calculated by the haptic rendering processor 142 and applies the calculated force to the user 146. The force
10 actuator 148 may be any device known in the art for applying a force to an object.

In one embodiment the haptic rendering processor 142 and the display processor 144 are different processors. In another embodiment, the haptic rendering processor 142 and the display processor 144 are the same processor. In yet another embodiment, the module generating a representation of an object in graphic space, the module
15 determining the user's haptic interface in graphic space, the module determining the user's fiducial object in graphic space, and the module calculating a force to be applied to the user in real space are separate devices.

Having described preferred embodiments of the invention, it will now become apparent to one of skill in the art that other embodiments incorporating the concepts may
20 be used. It is felt, therefore, that these embodiments should not be limited to disclosed embodiments but rather should be limited only by the spirit and scope of the following claims.